IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 20, 2020, accepted October 28, 2020, date of publication November 5, 2020, date of current version November 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3036037

The Application of Machine Learning in Self-Adaptive Systems: A Systematic Literature Review

THERESIA RATIH DEWI SAPUTRI [1] AND SEOK-WON LEE [2,3], (Senior Member, IEEE)
[1]Department of Informatics, Ciputra University, Surabaya 67219, Indonesia
[2]Department of Software and Computer Engineering, Ajou University, Suwon 443749, South Korea
[3]Department of Artificial Intelligence, Ajou University, Suwon 443749, South Korea

Corresponding author: Seok-Won Lee (leesw@ajou.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) Funded by the Ministry of Science and ICT under Grant NRF-2020R1F1A1075605.

ABSTRACT *Context*: Self-adaptive systems have been studied in software engineering over the past few decades attempting to address challenges within the field. There is a continuous significant need to fully understand the behavior and characteristics of the systems that operate in dynamic environments. By learning the behavior pattern of the environment, we can avoid unnecessary adaptations imbalance efforts for adaptation. As such, there exist research in the area of machine learning aimed at understanding dynamic environments regarding self-adaptive systems. *Objective*: This study aims to help software practitioners to address adaptation concerns by performing a systematic literature review that provides a comprehensive overview of using machine learning (ML) in self-adaptive systems. We summarize state-of-the-art Of the ML approaches used to handle self-adaptation to help software engineers in the proper selection of ML techniques based on the adaptation concern. *Method*: This review examines research published between 2001 and 2019 on ML implementation in self-adaptive systems, focusing on the adaptation aspects and purposes. The review was conducted by analyzing major scientific databases that resulted in 78 primary studies from 315 papers from an automatic search. *Result*: Finally, this study recommends three future research directions to enhance the application of machine learning in self-adaptive systems.

INDEX TERMS Systematic literature review, self-adaptive systems, machine learning, adaptation.

I. INTRODUCTION

Advancing technology and increasing user expectations lead to changing environments in software system development. Moreover, due to the increasing of the complexity software system should become more flexible, dependable, energy-efficient, recoverable, customizable, configurable, and self-optimizing by adapting to the changes. These changing environments in the software system development require human supervision to consistently maintain operations in all conditions. This maintenance can be both costly and time-consuming during operation. Therefore, Self-Adaptive Systems (SAS), systems that can adjust operations based on environmental conditions, are needed to achieve system goals. These goals should be able to address existing challenges in operations, including managing complexities and handling changing conditions. SAS are able to change or modify its configuration based on the changes in its environments through autonomous adaptation. Krupitzer *et.al.* [55] proposed a taxonomy of self-adaptation in the dimension of reason, time, technique, adaptation control, and level. This taxonomy is supported by more recent studies. Yuan *et.al.* [105] proposed a taxonomy of SAS especially focusing on self-protection properties including lifecycle focus, decision-making level, and response timing. Another work that provides a taxonomy for SAS is proposed by Claudia Raibulet [106]. This study includes some dimensions which are provided by Krupitzer *et.al.* such as reactivity and time.

The key idea of self-adaptation in SAS is to adjust artifacts or attributes in response to any changes within the context [76]. SAS is obligated to deal with run-time changes. There is a significant need to fully understand the behavior and characteristics of the system that operate in a dynamic

The associate editor coordinating the review of this manuscript and approving it for publication was Porfirio Tramontana.

environment. In order to develop an appropriate self-adaptive system, understanding the nature of the system will aid in determining the process change and factors affecting the change [23]. The SAS behavior implies that a certain development and change activities are shifted from development-time to run-time, while reassigning the responsibility for these activities from software engineer to the system itself [104]. The existing SAS frameworks require the engineer to construct and utilize a complex analytical model in which requires much effort [26]. Moreover, the analysis for optimal SAS configuration is computationally expensive, which can hurt the efficiency of SAS requiring prompt reaction to the situation at run-time. One of the main challenges in SAS is uncertainty. Weyns in [88] stated SAS should increase its trustworthiness by constructing a system that can operate under uncertain conditions. The work in [28] argues that one of the causes of uncertainty is due to a lack of knowledge surrounding the adaptation decision.

Various past research addresses the uncertainty in SAS. However, most of the works focused on acquiring direct values from the environments. This could result in an unnecessary adaptation with imbalance efforts for adaptation. Therefore, further analysis to understand behavior patterns is needed. This could support reactive adaptations by learning the pattern of the environments. The need beyond reactive adaptation requires SAS to support the analysis that addresses the environment's uncertainty.

Machine Learning (ML), one of the sub-fields of artificial intelligence, can overcome those difficulties by building computer programs that improve their performance by learning from some experience. ML can improve the adaptation process by understanding the environment pattern, so that the adaptation mechanism does not behave like a control mode. The study conducted by Ding *et. al.* in [24] shows a learning component can collaborate to make adaptation decisions while the system is running.

Various studies have been published in the area of ML regarding SAS [21], [35], [36]. However, these studies are scattered in different journals, conference proceedings, and research communities. While there are several experience reports on ML in SAS, the growing body of research is still missing. There are no systematic studies that have been performed in analyzing the ML application in SAS. As a result, there is no clear outlook on how ML contributes to helping in addressing the challenge in SAS. With this knowledge, we can get a comprehensive understanding of applying ML in a SAS setting.

We perform a Systematic Literature Review (SLR) of ML for SAS focusing on the topic of adaptation concerns, purpose, and model selections. It is important to understand the characteristics and adaptation problem before applying a ML technique. Therefore, we aim to identify the trends of applying ML in current engineering for SAS, assess the method for choosing specific ML techniques to address the problem of SAS, determine the limitations and success factors in the current approaches, and identify potential research directions for future work. The main objective of this study is to summarize the state-of-the-art of the ML implementation approaches in handling self-adaptation to allow for proper ML technique selection based on the adaptation concern.

The paper is organized as follows. Section 2 discusses existing works related to ML and SAS. The research method, including the research questions and research protocol, is presented in Section 3. Section 4 presents the result of the survey, while Section 5 discusses the main research findings and future research directions of ML in SAS. The threats of validity are explained in Section 6. Section 7 provides concluding remarks on this study.

II. RELATED WORKS

Since there is an increasing number of conducted studies in regard to SAS, there are various systematic literature reviews in this area. Kruptizer *et.al.* in [54] conducted a literature comparison by demonstrating different aspects of each approach including their types, support, and applicability. Even though the authors mentioned the comparison of the selected approaches cannot explicitly draw a conclusion about the software development process, we still can see the strengths and weaknesses of each approach.

Yang *et al.* in [96] investigate studies of requirements modeling and analysis for SAS. One of the challenges mentioned in this study is retroactive control mechanisms in existing SAS. Retroactive control mechanisms measure error and maintain the output so for the desired condition. Therefore, this literature review suggests the use of a feedforward-feedback control mechanism can tune the systems' behavior based on a measured disturbance at run-time.

Kruptizer *et al.* in [55] present a taxonomy of self-adaptation and surveys in engineering SAS. The proposed taxonomy focuses on time, reason, technique, level, and adaptation control. By analyzing these categories, they found the challenges in SAS involving the integration of the context and the proactivity adaption should be addressed to optimize the potential of context adaptation. One of the ways to address these challenges is to implement a multi-agent system. A multi-agent system is autonomous, reactive, and proactivity, which enhance the adaptation mechanism [45], [47].

Ye *et al.* in [97] provide a survey on the self-organization mechanism in a multi-agent system. One of the key findings in this study is reinforcement learning. This illustrates that the system has better scalability and is more suitable for real applications when the agents are able to predict other's policies. By learning about environment patterns, ML is able to help the decision-making process. ML, as such, has been used in different SE processes, including software vulnerability detection [38], cost estimation [41], and requirements analysis [57]. In SE-specific processes for SAS, ML has been utilized for improving environment understanding to adjust control of run-time behavior.

From the findings of various literature reviews in SAS, we can see that ML plays a significant role in enhancing SAS performance. There are various ways of evaluating the performance of a SAS. For example, an SAS can be evaluated based on its failure density, cost, or time to adapt. Raibulet *et al.* proposed a taxonomy for evaluating SAS [107] which includes evaluation scope, evaluation time, evaluation mechanism, evaluation perspective, and evaluation type. The study proposed by Chen *et al.* [17] shows that using a genetic algorithm for predicting Quality of Service (QoS) provided higher accuracy results. The aforementioned works focus on the general development of SAS. There is no evidence of studies that focus on analyzing the application of ML for enhancing the ability of SAS. The studies on ML for SAS are scattered between different journals and conferences.

Therefore, in this article, we present a systematic literature review to evaluate the implementation of proposed ML techniques in various adaptation concerns of SAS. Our study is focused on providing the engineer with a guide to select appropriate ML techniques based on the required adaptation needs, including modeling, reasoning, and validation. This selection could lead to improvements in giving proactive adaptations, rather than retroactive adaptation. Moreover, it is important to understand the application of machine learning in control loop due to the characteristic self-adaptive software system. SAS should handle the increasing complexity of managing software systems. Therefore, it should be deal with internal dynamic and dynamic environments. In order to meet with these requirements, self-adaptive should have an adaptation logic to manage the system.

III. RESEARCH METHOD

This study followed the principles of SLR proposed by Brereton *et al.* in [10].

A. RESEARCH QUESTIONS

The main objective of this study was to study and summarize the state-of-the-art implementation of machine learning approaches to address adaptation concern by identifying relevant mature study in SAS. Therefore, to address the research gap, we conducted a systematic literature review to address the following research questions:

- RQ1. What are the trends in ML research concerning SAS over the last 19 years?
  Rationale: This question examined ML approaches to address adaptation challenges in SAS and compares it with other modeling techniques.
- RQ2. What ML techniques are researchers and practitioners mostly working on and in what manner are ML techniques utilized to address adaptation concerns in SAS?
  Rationale: This question was intended to classify machine learning techniques that have been used. From this question, we also aimed to identify how the technique is used, and whether it is for modeling, reasoning, or modeling checking. This question was also used to analyze the implementation of machine learning techniques in adaptation processes including its time aspect. The process was classified into five concerns, including verification, model, framework, behavior, and architecture.
- RQ3. What approaches are used to select ML techniques and how appropriate are these techniques in addressing the adaptation concerns in a specific domain application?
  Rationale: This question was intended to identify reasons for selecting a particular machine learning technique. The reason was classified into no justification, explanatory analysis, and direct comparison (with other approaches).
- RQ4. What are the challenges and success factors to apply ML in SAS?
  Rationale: This question was used to understand limitations of using ML for adaptation and identify potential research directions in the use of ML in SAS.

RQ1 was used to understand trends in using machine learning approaches to address adaptation challenges in SAS and how often the self-* property has been addressed. The term "how often" refers to the number of papers which use machine learning to address self-* properties such as self-adaptation, self-configuration, self-healing, self-learning, self-management, and self-optimization. RQ2 was intended to classify the machine learning techniques that have been used. We also aimed to analyze the implementation of machine learning techniques in the adaptation process. The process was classified into five concerns include verification, model, framework, behavior, and architecture. RQ3 was intended to find the reason for choosing a machine learning technique. From this question, we aimed to identify how the technique is used, whether it is for modeling, reasoning, or modeling checking. From this question, we also aimed for identifying the existing approach for selecting machine learning approach. The answer was classified into no justification, explanatory analysis, and direct comparison (with other approaches). RQ4 was used to get the insight in the limitation and success factor of using machine learning for adaptation. This result was used to identify potential research direction in the use of machine learning is SAS.

B. RESEARCH PROCESS

The procedure for this systematic literature review consisted of five main steps as seen in Figure 1. The scope of the search was within widely known science directories, journal, and conferences in the related field. The data search was

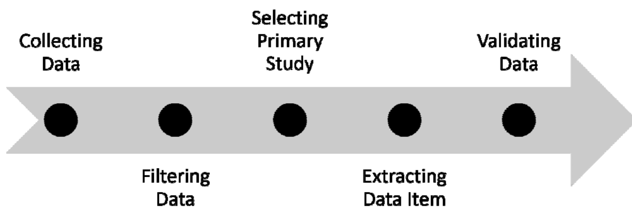

FIGURE 1. Research Method for Conducting SLR.

performed by searching the data automatically from electronic data sources (DS). The gathered papers were filtered based on their abstract and full text using the defined inclusion and exclusion criteria. The data from selected primary studies were extracted and synthesis to answer the research questions. Then, the data items were defined as the evidence to answer the research questions. In order to address the threats of validation, we performed the cross-check analysis between researchers to review the developed research protocols. Those researchers were asked to check the search scope, data item, and evaluation criteria.

1) SEARCH STRATEGY

The search strategy was important in a systematic literature review to ensure the completeness of the conducted study. The automatic search was used to retrieve the paper in a number of selected electronic databases. The automatic searches were performed in five scientific databases as listed in Table 1. The study conducted by Chen *et al.* in [13] analyzed the ten most-used DS by SE researchers. We select top five data sources to reduce the amount of redundant data.

TABLE 1. The scope of automatic searching.

INDEX	NAME	URL
DS1	ACM Digital Library	http://dl.acm.org/
DS2	IEEE Xplore Digital Library	http://ieeexplore.ieee.org/
DS3	ScienceDirect	http://www.sciencedirect.com/
DS4	SpringerLink	https://link.springer.com/
DS5	Wiley InterScience	https://onlinelibrary.wiley.com/

To collect the data with an automatic search function, we used specific keywords from three categories related to machine learning, self-adaptiveness, and software identified in the papers' metadata, including the title and abstract. Table 2 list the categories and the keywords on matching topics. The machine learning category was used to search any paper which use machine learning approaches in their study.

TABLE 2. Category and related search terms used.

CATEGORY	KEYWORDS
Machine Learning	'machine learning', 'learning', 'neural network', 'support vector machine', 'naive bayes', 'decision tree', 'fuzzy', 'cluster', 'case-based', 'reinforcement learning', 'intelligence', 'deep'
Self-adaptiveness	'adaptive', 'adaptation', 'self', 'autonomous', 'autonomic', 'automated', 'dynamic', 'aware'
Only Software Related	'software', 'system', 'run-time'

The purpose for including specific machine learning techniques in the keywords was to gather studies that do not specifically mention 'machine learning' but use machine learning technique in their approach. Since there were various manuscripts about machine learning in the listed DS, we also used specific keywords in self-adaptive software. The primary study should provide insight on the machine learning application in SAS. The purpose of using 'only software related' keywords was to avoid any paper from non-SE fields such as mobile communication, network, and specific algorithm optimization.

In order to avoid any potential missing of relevant studies, we also extended our automatic search in Google Scholars. We checked against top 1000 result search string "machine learning in self-adaptive" with the publication range from 2001 to 2019.

2) STUDY SELECTION

In order to select the primary study, we conducted two rounds of study selection against the results from automatic searching. In the first round, the researchers should filter the papers based on the title and abstract manually. In this round, the researchers should remove duplicate data from selected data sources and Google Scholar. In the second round, we applied the following well-defined inclusion and exclusion criteria to filter the primary studies.

- Inclusion Criteria 1: The date of publication is between January 2001 - December 2019.
  Rationale: The research uses 2001 as the starting year because SAS studies were actively researched around that time.
- Inclusion criterion 2: The study proposes a machine learning technique for the system as well as the property. However, any studies which combine learning algorithm with the formal method such as Markov chain model or Fuzzy logic will be included.
  Rationale: This study includes every study that proposed a methodology or at least a model using machine learning.
- Inclusion criterion 3: The study focuses on adaptation logic concerns.
  Rationale: Since the study focuses on SAS, we only include studies that use machine learning for adaptation purpose.
- Exclusion criterion 1: The study should not be an editorial or abstract.
  Rationale: These types of studies are excluded since they usually provide limited information about the proposed approach. Also, editorial papers do not provide any approach formal method.
- Exclusion criterion 2: The study should not use a formal method alone (e.g.: state machine, automata, Markov model, or Petri net) rather than a machine learning technique.
  Rationale: This study does not answer the study question since we focus our review on machine learning algorithm.

TABLE 3. Data item collection form.

INDEX	DATA FIELDS	DESCRIPTION	RESPECTED RESEARCH QUESTION
IF01	Title	The title of the primary study	For documentation purpose
IF02	Year	The publication year of the primary study	RQ1
IF03	Self-* property	Adaptation, configuration, decision, healing, management, optimization or tuning	RQ1
IF04	Publication source	The name of publication venue and type of the primary study and its publisher	For documentation purpose
IF05	Machine learning approach	The machine learning technique used in the primary study	RQ2
IF06	Adaptation concern	Architecture, behavior, framework, model or verification	RQ2
IF07	Time for Adaptation	Proactive adaptation, reactive adaptation	RQ2
IF08	Purpose of use	Modeling, reasoning or model checking	RQ3
IF09	Machine learning selection method	Theoretical analysis, direct comparison, or no justification	RQ3
IF10	Assessment method	Case study, experimental study, no evaluation	RQ3
IF11	Strength	The benefits of using machine learning that have been identified in the study	RQ4
IF12	Limitation	The challenges of using machine learning that have been identified in stud	RQ4

TABLE 4. Summary of conducted literature review.

STEP IN RESEARCH METHOD	# OF STUDIES	REMARK
Studies that were filtered based on the title and abstract	315	Not Available
Studies that were filtered based on inclusion and exclusion criteria	112	Not Available
Studies that were selected as primary study by main researchers	80	by applying validation 1, 32 studies were rejected
Studies that were selected as primary study by independent researchers	75	by applying validation 1, 37 studies were rejected
Studies that were selected as final primary studies	78	two papers were removed after discussion session

- Exclusion criterion 3: The study that focus on the level of communication such as network infrastructure and communication structure rather than application and system software.
  Rationale: This study does not answer the study question since we focus our review in machine learning application for self-adaption software systems.

3) DATA EXTRACTION

The scope of the search process was limited to the indicated scientific databases, journals and conferences. Each paper was evaluated based on the extracted data item (IF) listed in Table 3. The data items were defined as evidence that will provide answers to the research questions. The data items were extracted manually from the papers collected, using both manual search and automatic search functions.

The data item title (IF01) was used for documentation purposes. Year (IF02) is used for answering RQ1. By using this data item, we also intended to find the trend of using machine learning approaches. We also identified the self-* property (IF03) in SAS. The self-* property referred to the capability of a software system in term of its ability to meet with the environment changes. The publication source (IF04) was used to obtain a better understanding in the trends of using machine learning techniques in SAS.

Machine learning technique (IF05) was used to answer RQ2. This question was intended to determine the application of machine learning technique. Adaptation concern (IF06) data field referred to the general subject of SAS study. The category of the concern of adaptation was adopted from a survey conducted by Weyns *et al.* in [89]. Meanwhile time for adaptation (IF07) data field referred to the question when the system should perform adaptation. We adopted the taxonomy related to time from Kruptizer *et al.* in [55] divided into proactive and reactive.

Next, purpose of use (IF08), machine learning selected method (IF09) and assessment method (IF10) were defined to answer for RQ3. Those data items were used to analyze how the machine learning is chosen and evaluated.

Finally, the strength (IF11) and limitation (IF12) were used to answer RQ4. Those data items were collected to identify the contribution and limitation in applying machine learning for adaptation. Identifying those data items is important to understand the future research direction.

4) DATA VALIDATION

In order to address threats of validity related to the data, we performed two validation methods for the filtered studies, validation 1 (cross-checking agreement) and validation 2 (independent expert checking). The result of each step was summarized in Table 4.

Validation 1 evaluated the consistency of extracted data item from the 112 filtered studies. This validation was done

TABLE 5. Evaluation criteria form.

INDEX	EVALUATION CRITERIA	ANSWER OPTION AND SCORING
QC1	How does the author(s) explain their research problem?	– The author(s) provides explicit problem description = 2 – The author(s) provides general problem description = 1 – There is no problem description = 0
QC2	How does the author(s) present their research design?	– The author(s) gives explicit explanation for the study plan or proposed approach = 2 – The author(s) provides general description for the study plan or proposed approach = 1 – There is no description for the study plan or proposed approach = 0
QC3	Does the paper have comprehensive validation method?	– The author(s) applies scientific validation method for example empirical validation = 2 – The author(s) only briefly explains the validation method = 1 – There is no description for the validation method = 0
QC4	How does the author(s) present study limitation?	– The author(s) provides explicit list of study limitation = 2 – The author(s) only briefly explains the study limitation= 1 – There is no description for the study limitation = 0

individually by the main researcher by checking whether the selected studies were able to answer the research questions. Once the main researchers finished the validation process, they exchanged results and analyze the result of their partners. By combining the final result for validation 1, the main researcher agreed to reject 32 studies due to their lack of ability for answering the research questions. The second validation was conducted by independent researchers which are a PhD candidate in computer engineering and a Postdoc researcher working in Smart Grid System.

In order to avoid bias, two independent experts were asked to conduct the second validation. This validation was performed on the 112 filtered studies. From this activity, 37 studies were rejected by experts.

A discussion session is conducted to match the result of validation 1 and validation 2. We found that there is a high percentage of agreement (91% agreed primary studies) between the main researchers and independent experts. There were some disagreements on the selected primary studies. Therefore, those papers were carefully analyzed against the research questions. Finally, two papers were removed from the list of primary study. As a result of the validation process, 78 papers were selected as the primary studies in this work as seen in Appendix Table 14. Those 78 primary studies were selected from various journals and conferences in 54 different publication venues seen in Appendix Table 15. Most of the primary studies (60 out of 78) are published in self-adaptive related conferences.

5) ASSESSMENT OF PRESENTATION QUALITY

The quality criteria (QC) were identified to determine the quality of the primary studies. Checking the quality of the data was important for the further analysis specially to identify the limitation and challenges of the studies. In order to assess the quality, we collected a set of quality criteria by adopting the criteria proposed by Dyba and Dingsoyr in [25]. Each of the primary studies was evaluated based on the quality assessment score (maximum 8) calculated by summing up the scores for the entire questions for a study. The scores were varied based on the option seen in Table 5.

Figure 2 shows the distribution of the primary studies over the quality criteria. More than 50% of the primary studies described those quality criteria explicitly. It indicated that most of the primary studies are high quality sources of data in terms of defining the research problem, describing the research design and the proposed approach, and determining the validation method.

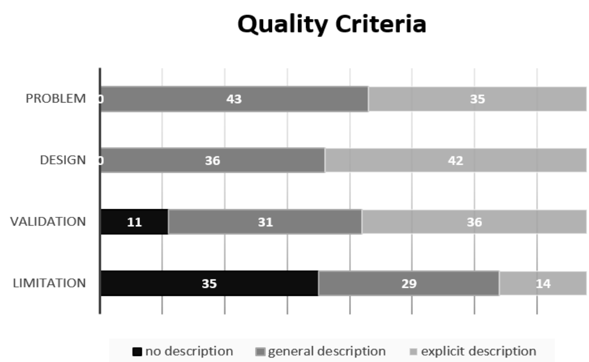

FIGURE 2. Primary studies quality checking.

However, the score for describing the limitations was very low. 35 out of 78 papers do not provide descriptions of the limitations. Only 14 primary studies gave explicit descriptions in regard to the limitations of the study. The rest of the papers only provided limited descriptions about the limitations.

The study shows that the average of the total score is 5.09 out of 8. This means that most of the primary studies in terms of quality are mediocre, neither perfect nor completely lacked. In order to improve the quality, the researchers should put more attention when reporting about the limitation of the self-adaptation. Finding the limitation of the work is

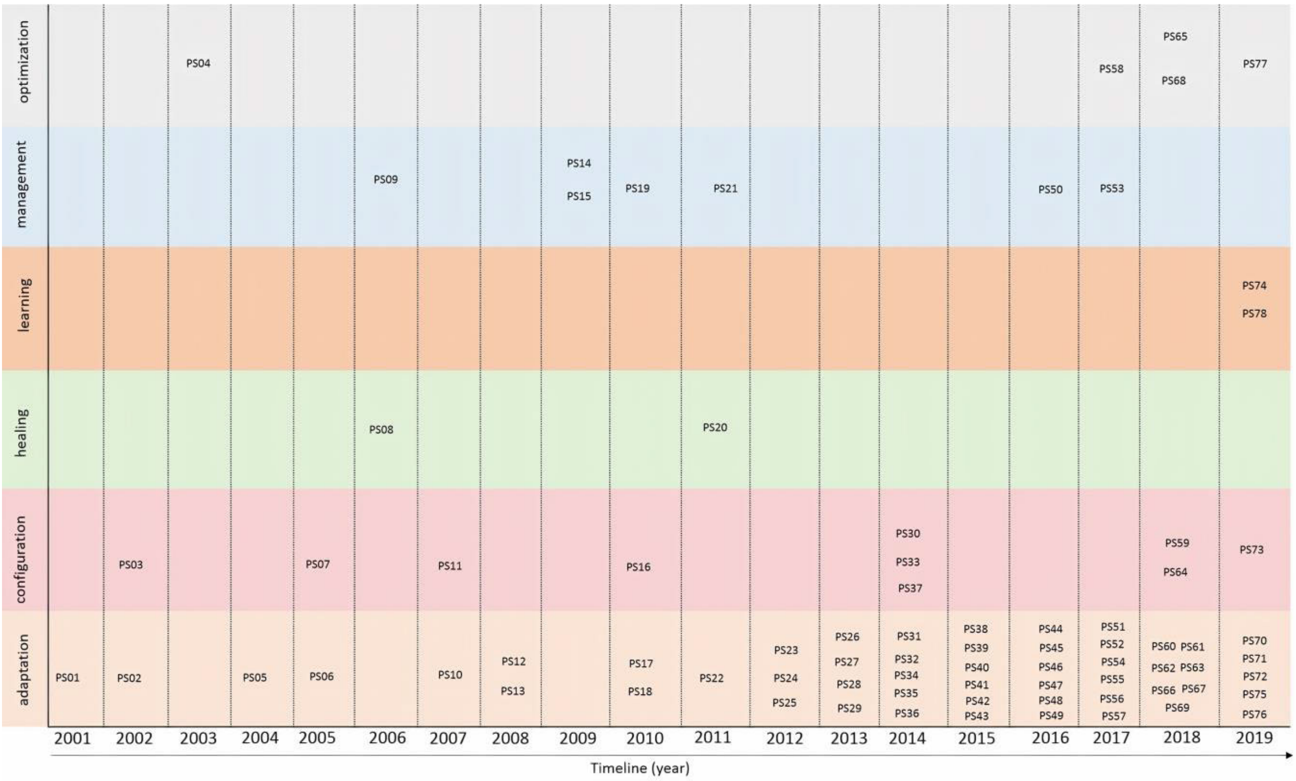

FIGURE 3. Trend of using machine learning technique in SAS based on its self-* properties.

important to ameliorate the proposed approach completely insufficient for future research.

IV. RESULT AND ANALYSIS

We performed this systematic literature review according to the procedure described in Section 3. We present the results by answering each research question using the extracted data item collection.

A. RQ1. WHAT ARE THE TRENDS IN ML RESEARCH CONCERNING SAS OVER THE LAST 19 YEARS?

Various studies argue that the use of formal methods in developing self-adaptive systems provides mechanisms to avoid undesirable behavior [90]. This study shows that most of the proposed models use a hard-coded method for adaptation logic, which can lead to the inflexibility. Machine learning can address this problem by determining the pattern and predicting the behavior. However, little attention is given to a machine learning approach.

Figure 3 shows distribution of the primary studies based on the trends derived from the year data field (IF02). The data are gathered after applying the second exclusion criteria. Even though the numbers are small, there is a remarkable progression in the number of studies that use machine learning. Particularly, since 2014, there are at least six studies conducted per year—a sharp increasing trend in comparison to previous years.

Many SAS are described with various terms, such as ‘‘self-healing’’, ‘‘self-configuration ’’ and ‘‘self-organizing’’, to maintain its complexity. Those terms are usually labeled ‘‘self-*’’ properties. We extracted the ‘‘self-*’’ property based on the category discussed by Bern and Ghosh in [8]. If the ‘‘self-*’’ is not described explicitly, we categorized it as ‘‘self-adaptation’’.

Based on the extracted data field in terms of the ‘‘self-*’’ property, we can see that machine learning mostly fell in self-adaptation (52 out of 78), and self-configuration or self-organization (ten out of 78). The machine learning application in other properties such as self-healing, self-management, and self-optimization are introduced later in 2003. Meanwhile, the concept of self-learning is introduced in 2019 with the ability to support continuous learning. By having this ability, a SAS is able to perform runtime model verification.

B. RQ2. WHAT ML TECHNIQUES ARE RESEARCHERS AND PRACTITIONERS MOSTLY WORKING ON AND IN WHAT MANNER ARE ML TECHNIQUES UTILIZED TO ADDRESS ADAPTATION CONCERNS IN SAS?

The answers to RQ2 are derived from the data item ML approach (IF05), Adaptation Concern (IF06), and Time for Adaptation (IF07). This question is intended to determine the application of ML technique.

The first data item is used to identify what kind of ML technique usually used for handling adaptation. Each of the techniques has unique characteristics based on the task and their needs of knowledge. These machine learning types are chosen based on the common taxonomy in machine learning algorithm such as supervised learning, unsupervised learning, and reinforcement learning. During our study selection we found that there were some studies which apply deep learning algorithm, one of machine learning algorithms, for supporting adaptation process. However, this algorithm was not included in this study due to its domain application. All of the studies for SAS using deep learning published between 2000 and 20019 focused on network communication. Therefore, these studies were filtered out by the third exclusion criteria in our study.

Table 6 shows distribution of primary studies over the various ML techniques used in SAS. We found there are eight types of ML techniques used to address adaptation. The result shows that reinforcement learning is a top ML technique used in adaptation [43], [58], [65], [93]. Reinforcement learning is a semi machine learning algorithm which use the concept of Markov Decision Program to understand the uncertainty in the environment. Therefore, Reinforcement learning use information of the state in the environment to decide the action so that it can meet with the requirements environment condition. It combines the Markov chain model with reward-policy mechanism. This is one of the reasons why Markov model alone is not included in this study because it is not a machine learning algorithm. Reinforcement learning technique is used due to its ability to provide multi-objective decision making using the reward-punishment mechanism. The reinforcement learning handle uncertainty in the environment by having a set of solutions through dynamic feedback interaction with the environment. This ability is important in SAS because we can address the uncertainty situations by adjusting the policies based on the new conditions. Another advantage of reinforcement learning is the possibility to extend the algorithm based on its existing domain knowledge and requirements.

The second most-used ML technique is fuzzy learning. We categorized a study in fuzzy learning when it combined any machine learning algorithm with fuzzy logic. The fuzzy logic is very beneficial for SAS studies that are vague and or uncertain. Fuzzy logic is not an algorithm [64] but a value logic. Therefore, in this work, the term 'fuzzy learning' means a learning algorithm that uses fuzzy logic. Various studies also used a neural network in SAS [67], [86].

Other studies use Bayesian Theory, Decision Tree and Clustering. The work proposed by Chen *et al.* in [12] used decision tree to address adaptive architectural design decisions to find the best design for the desired requirements within the contextualized solution space. Meanwhile, the work proposed by Lu and Cukic in [62] used naïve bayes as a base learner to support adaptive software failure adaptation. This study shows that naïve bayes can efficiently adapts fault prediction to the dynamics of software development in which modules are developed over time.

We investigated the trends further by looking for the distribution for over the years. We can see in Figure 4 that there is a consistent use of reinforcement learning and fuzzy learning from the beginning until currently. There was an increasing use of neural network in the recent years. We argue this phenomenon occurred due to the increasing advance research on neural network such as deep learning.

TABLE 6. Machine learning technique used in SAS.

ID	NAME	# of PS	PRIMARY STUDIES
1	Bayesian Theory	7	PS01, PS23, PS27, PS41, PS57, PS67, PS75
2	Clustering	4	PS15, PS31, PS45, PS52
3	Decision Tree	8	PS24, PS26, PS36, PS40, PS60, PS65, PS68, PS73
4	Fuzzy Learning	17	PS03, PS05, PS06, PS08, PS11, PS13, PS16, PS17, PS18, PS20, PS22, PS32, PS34, PS35, PS46, PS48, PS50
5	Genetic Algorithm	10	PS10, PS21, PS28, PS30, PS33, PS37, PS43, PS44, PS66, PS78
6	Neural Network	11	PS04, PS07, PS19, PS29, PS47, PS53, PS55, PS56, PS58, PS61, PS77
7	Regression	2	PS71, PS72
8	Reinforcement Learning	19	PS02, PS09, PS12, PS14, PS25, PS38, PS39, PS42, PS49, PS51, PS54, PS59, PS62, PS63, PS64, PS69, PS70, PS74, PS76

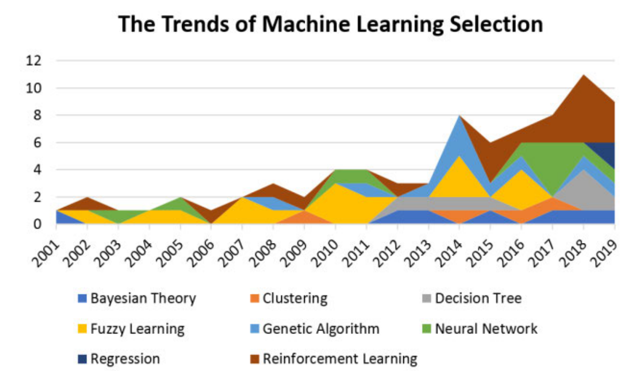

FIGURE 4. Trend of selecting machine learning techniques over the years.

It is worth noticing that combining different ML can significantly improve performance. This align with the high number of primary studies which apply fuzzy learning for adaptation in SAS. A well-known combination is the Fuzzy-Neural network with self-learning abilities that can adapt to meaningful change. The studies in [36] and [22] argue the method can handle dynamic uncertainties by sensing the environments. It has the ability to update self-adaptation logic with the interference of software engineers with limited knowledge of fuzzy control.

The next data item, concern of the subject data field, refers to the general subject of the study. The category of the concern of subject is adopted from a survey proposed by Weyns *et al.* in [89]. Table 7 shows a summary of the concerns in the adaptation addressed using ML approaches. For this data item, behavior concerns that focused on the behavior adaptation of a software system identified 27 studies. We also found that 26 primary studies used ML for model concerns. The model varies from the environment model to the software model. This concern is related to modeling tasks for SAS. This concern is followed by the architectural concern, with16 studies. In those studies, ML techniques are used for managing the artifacts in software system. The behavior concern focuses on any aspect of behavior for adaptation.

TABLE 7. Concern objects for adaptation in SAS.

ID	NAME	# of PS	PRIMARY STUDIES
1	Architecture	16	PS07, PS09, PS14, PS16, PS17, PS 28, PS30, PS32, PS41, PS45, PS55, PS64, PS68, PS70, PS72, PS74
2	Behavior	27	PS01, PS02, PS10, PS13, PS18, PS19, PS20, PS34, PS35, PS39, PS40, PS43, PS44, PS47, PS49, PS51, PS52, PS54, PS57, PS58, PS61, PS62, PS67, PS69, PS71, PS77, PS78
3	Framework	5	PS04, PS06, PS08, PS48, PS50
4	Model	26	PS03, PS05, PS11, PS12, PS15, PS21, PS22, PS24, PS25, PS26, PS27, PS29, PS31, PS33, PS36, PS38, PS42, PS46, PS53, PS56, PS59, PS63, PS65, PS66, PS73, PS75
5	Verification	4	PS23, PS37, PS60, PS76

We found that reinforcement learning is usually used for model and behavior adaptation. Meanwhile for architecture and framework adaptations, most of these studies prefer to use fuzzy learning. Four papers used ML for verification concerns. These studies focused on verification tasks, including runtime verification and model verification. Most of these works used ML for model checking. The work in [31] uses Genetic Algorithm to verify the model at runtime in response to changing system and environmental condition.

The work in [18] discusses two common characteristics in SAS. The first characteristic is that a decision should be made in the runtime. The second characteristic is the system should reason their state in the environments in the runtime. Continuous monitoring environment in SAS is important to provide required adaptation so that the system can meet with required behavior based on environment condition. Therefore, a mechanism to control dynamic behaviors in SAS is needed. The control mechanism plays an important role because SAS should continuously observe the non-controllable environment using adaptation logic. The adaptation process implements a control loop in line with the monitor-analyze-execute-execute-knowledge (MAPE-K) [48] loop. It is important to understand the application of ML in a MAPE-K loop due to the characteristics of SAS. It should be able to handle the increasing complexity of managing software systems. Therefore, SAS should deal with internal dynamic and dynamic environments. In order to meet with these requirements, SAS should have an adaptation logic to manage the system.

To improve the self-adaptation, understanding the process inside the control loop, the core process of SAS, is necessary. The control loop starts with monitor activity. This activity is done to collect relevant data in the environment that reflects the current state of the system. Next, the system should analyze the result of monitor activity. There are many approaches that can be used to structure and reason that information. One of the approaches taken to improve the self-adaptation process is implementing ML technique in analyzing activity. One of the goals of ML is making the system actively learn about the existing knowledge by getting complete understanding of the current state of the system. Therefore, implementing ML is important to fully comprehend the current state of the system.

Without ML techniques, the adaptation processes in SAS behave like a control mode. Instead of the system learning to adapt, the system changes into different configuration that have been constructed before. There are various control loops used in SAS including the control loop in the area of control engineering and autonomous feedback control loop [108]. Table 8 shows the distribution of primary studies regarding control loop methods.

TABLE 8. Machine learning in control loop method for SAS.

ID	NAME	# of PS	PRIMARY STUDIES
1	MAPE	1	PS43
2	MAPE-K	22	PS15, PS27, PS28, PS30, PS30, PS31, PS33, PS36, PS37, PS38, PS41, PS42, PS49, PS51, PS59, PS60, PS61, PS63, PS64, PS65, PS68, PS71, PS72
3	No MAPE concept	55	PS01, PS02, PS03, PS04, PS05, PS06, PS07, PS08, PS09, PS10, PS11, PS12, PS13, PS14, PS16, PS17, PS18, PS19, PS20, PS21, PS22, PS23, PS24, PS25, PS26, PS29, PS32, PS34, PS35, PS39, PS40, PS44, PS45, PS46, PS47, PS48, PS50, PS52, PS53, PS54, PS55, PS56, PS57, PS58, PS62, PS66, PS67, PS69, PS70, PS73, PS74, PS75, PS76, PS77, PS78

We investigated whether primary studies consider the implementation of MAPE-K loop [20] for addressing adaptation mechanism. Most of the primary studies (55 out of 78) used a traditional control loop. The traditional control loop refers to the autonomous feedback control loop which include collect, analyze, decide, and act [110]. More than 20% of the primary studies use MAPE concept for the adaptation process. Among the primary studies, 22 of them implement the full MAPE-K concepts. Only one (1) primary study uses ML in the MAPE adaptation. The work in [30] proposed a method called AutoRELAX, which uses genetic algorithms and fuzzy logic functions in their adaptation process. The author argues that implementing ML techniques affects the abilities of an adaptive system to satisfy not only the requirements in the presence of system but also the environmental uncertainty. The work in [7] shows how dynamic Bayesian networks can be used to enhance the decision-making process. It shows how the decision can correspond with changes in the environments, such as network link failure and unreliable monitoring data.

Kruptizer *et.al.* in [55] argued that it is important to understand the time level for adaptation. Table 9 shows the summary of supported time for adaptation in SAS. Ideally, having a proactive adaptation is preferable compare to reactive adaptation so that there is no interruption in the performance. However, proactive adaptation requires more accurate prediction ability to support continues monitoring and learning. A SAS is categorized as reactive system when the adaptation start after there is a need for change. Meanwhile, a SAS is categorized as proactive system when the adaptation happened before the performance dropped due to the change in the environments.

TABLE 9. Time for adaptation in SAS.

ID	NAME	# of PS	PRIMARY STUDIES
1	Proactive	36	PS01,PS05,PS07,PS08,PS12,PS14, PS18,PS19,PS20,PS23,PS24,PS27, PS29,PS30,PS31,PS33,PS35,PS36, PS40,PS42,PS54,PS56,PS58,PS60, PS62,PS64,PS65,PS66,PS67,PS68, PS69,PS73,PS74,PS76,PS77,PS78
2	Reactive	42	PS02,PS03,PS04,PS06,PS09,PS10, PS11,PS13,PS15,PS16,PS17,PS21, PS22,PS25,PS26,PS28,PS32,PS34, PS37,PS38,PS39,PS41,PS43,PS44, PS45,PS46,PS47,PS48,PS49,PS50, PS51,PS52,PS53,PS55,PS57,PS59, PS61,PS63,PS70,PS71,PS72,PS75

We found that more than 50% of the primary studies support reactive adaptation (42 out of 78). In reactive system, the adaptation process focuses on finding the unregular pattern from the environment so that the system can react based on the change [9], [30]. In proactive system, the adaptation process focuses on predicting the possible change in the environments [4], [75]. Interestingly, even though proactive and reactive systems have different approach in terms of time level, many of the study apply similar control loop.

C. RQ3. WHAT APPROACHES ARE USED TO SELECT ML TECHNIQUES AND HOW APPROPRIATE ARE THESE TECHNIQUES IN ADDRESSING THE ADAPTATION CONCERNS IN A SPECIFIC DOMAIN APPLICATION?

The answer to RQ3 is derived from the data items purpose of use (IF08), selection method (IF09), and assessment method (IF10).

We further investigated the purpose of implementing ML techniques in SAS. Table 10 shows the type of purpose that is used in the study (IF08). The majority of the primary studies (41 out of 78) shows that ML techniques were mostly utilized for a reasoning purpose [17], [49], [61], [93]. For a reasoning purpose, various ML techniques are used from Bayesian theory to genetic algorithms. Similar with the result presented in Table 6, most of the works (20 out of 41) applied reinforcement learning or fuzzy learning for a reasoning purpose. Most of the research used ML to reason regarding the design of SAS from the architecture design to software design.

TABLE 10. Purpose of using machine learning in SAS.

ID	NAME	# of PS	PRIMARY STUDIES
1	Model Checking	5	PS11, PS23, PS37, PS60, PS61
2	Modeling	32	PS01, PS03, PS04, PS05, PS08, PS09, PS12, PS20, PS22, PS24, PS25, PS26, PS28, PS29, PS31, PS33, PS34, PS36, PS40, PS42, PS47, PS57, PS58, PS9, PS66, PS71, PS72, PS73, PS74, PS75, PS76
3	Reasoning	41	PS02, PS06, PS07, PS10, PS13, PS14, PS15, PS16, PS17, PS18, PS19, PS21, PS27, PS30, PS32, PS35, PS38, PS39, PS41, PS43, PS44, PS45, PS48, PS49, PS50, PS52, PS53, PS54, PS55, PS56, PS62, PS63, PS64, PS65, PS67, PS68, PS69, PS70, PS77, PS78

At run-time, SAS is obligated to react based upon the situation. By using a ML approach, the system can determine the appropriate behavior based on the change environments. The work proposed by Gouin-Vallerand et. al in [34] argued the use of ML technique for reasoning supports the situation when there is no precise evaluation and knowledge. The algorithm provides reasoning rules, which do not need accurate knowledge of the model that can be difficult to obtain.

We further investigate the verification type of model verification approach. Table 11 summarize the model verification for adaptation in SAS. Most of the studies (49 out of 78) use static verification method in which done offline model building and verification. Meanwhile the rest of study applied dynamic verification method where the model built online. Rodrigues *et al.* in [72] proposed a system which apply online continues learning for adaptation. Having a dynamic model verification approach is more beneficial compare to static model verification due to its ability to provide more understanding in environment condition and behavior.

One of the key insights derived from the data item selection method (IF09) is that there is a lack of reason for choosing ML techniques. Table 12 shows that 24 out of 78 studies implement a particular ML technique in SAS without providing a specific justification. Only 17 out of 78 studies use experimental method to choose machine learning algorithm. By performing this procedure, they aim to understand the domain knowledge and find the algorithm with the best

TABLE 11. Model verification types for adaptation in SAS.

ID	NAME	# of PS	PRIMARY STUDIES
1	Static	49	PS03,PS04,PS05,PS07,PS08,PS09, PS11,PS13,PS15,PS16,PS18,PS19, PS20,PS21,PS22,PS27,PS28,PS29, PS32,PS34,PS35,PS36,PS37,PS39, PS42,PS43,PS44,PS45,PS46,PS47, PS48,PS49,PS50,PS53,PS54,PS55, PS56,PS57,PS58,PS65,PS66,PS68, PS69,PS70,PS72,PS73,PS76,PS77, PS78
3	Dynamic	29	PS01,PS02,PS06,PS10,PS12,PS14, PS17,PS23,PS24,PS25,PS26,PS30, PS31,PS33,PS38,PS40,PS41,PS51, PS52,PS59,PS60,PS61,PS62,PS63, PS64,PS67,PS71,PS74,PS75

TABLE 12. Justification procedure to choose machine learning in SAS.

ID	NAME	# of PS	PRIMARY STUDIES
1	Experiment Comparison	17	PS03, PS05, PS06, PS07, PS08, PS10, PS16, PS28, PS30, PS41, PS44, PS53, PS56, PS62, PS65, PS67, PS77
2	Theoretical Analysis	37	PS17, PS18, PS19, PS20, PS21, PS25, PS26, PS27, PS31, PS32, PS33, PS35, PS37, PS38, PS43, PS46, PS47, PS49, PS51, PS52, PS58, PS59, PS60, PS61, PS63, PS64, PS66, PS68, PS69, PS70, PS71, PS73, PS74, PS75, PS76, PS78
3	Without Justification	24	PS01, PS02, PS04, PS09, PS11, PS12, PS13, PS14, PS15, PS22, PS23, PS24, PS29, PS34, PS36, PS40, PS42, PS48, PS50, PS54, PS55, PS57, PS72

TABLE 13. Assessment method to validate the proposed machine learning technique.

ID	NAME	# of PS	PRIMARY STUDIES
1	Case Study	30	PS04, PS08, PS10, PS12, PS13, PS15, PS16, PS17, PS18, PS19, PS22, PS25, PS27, PS31, PS34, PS37, PS38, PS39, PS40, PS41, PS42, PS43, PS45, PS46, PS50, PS55, PS59, PS63, PS75
2	Experimental Study	41	PS01, PS03, PS05, PS06, PS07, PS09, PS11, PS14, PS20, PS21, PS23, PS24, PS26, PS28, PS29, PS30, PS32, PS33, PS35, PS36, PS44, PS51, PS52, PS53, PS56, PS58, PS60, PS61, PS62, PS64, PS65, PS66, PS67, PS68, PS69, PS70, PS71, PS72, PS73, PS77, PS78
3	Without Assessment	7	PS02, PS47, PS48, PS54, PS57, PS74, PS76

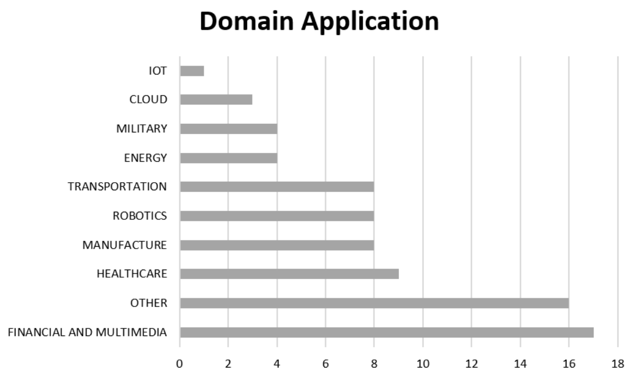

FIGURE 5. Application domain that use machine learning technique.

performance to learn from the data. Meanwhile, 37 out of 78 use theoretical analysis to choose the algorithm.

We also looked at the validation process used to evaluate the proposed method (IF10). This data item is used to determine whether the primary study provides adequate evaluation for the proposed method. Table 13 shows the summary of evaluation method on ML in SAS. The results indicate that there are two types of evaluation methods, experimental and case studies. The case study refers to a study in which the validation was performed in the application of proposed study in a single case. In this study, the evaluation is discussed with depth analysis. Meanwhile, the experimental study refers to a study which use more than one case to evaluate proposed approach.

Table 13 shows three categories because 7 studies only give a general description of the evaluation without providing further supportive evidence of the claim.

Further investigation is done in the domain application field. Figure 5 shows the distribution of primary studies over the application domain. The result shows that 79% of the studies use explicit domain application. However, the rest of the studies only provided a general domain in which could be considered as independent domain application.

The majority of the studies use financial and multimedia as their domain application (17 out of 78). These domain applications include cloud computing and social network domain applications. In the area of financial and multimedia, there are various ML techniques such as reinforcement learning, fuzzy logic, and genetic algorithm. Those algorithms are used for reasoning and modeling purpose. The second rank in the domain application healthcare domain in which the ML techniques are used for modeling, reasoning, and model checking. It is interesting to note that all the studies within the healthcare domain application used fuzzy logic to reason and model the architecture. We noted that some domain applications such as IOT and cloud have a limited number of studies which applied machine learning for adaptation. We found that the application of machine learning in these domains started in the recent years.

Different ML can be implemented in different stages due to varying characteristics of ML techniques. The authors rarely consider their own domain knowledge and the sufficiency of the data. In some cases, some researchers preferred usage of the fuzzy learning technique because of its ability to handle complex data, despite not having sufficient domain knowledge. The other factor that should be considered in selection of a ML technique are its characteristic. Naive Bayes technique has strong assumptions independently of the data. Without regard to the data dependency, the developer may select the Naive Bayes over other technique due to its simplicity. Many developers usually err in this regard because they lack an analysis of the relationship between the features. Therefore, one of the important factors to be considered is an understanding of the domain problem, including the domain knowledge.

D. RQ4. WHAT ARE THE CHALLENGES AND SUCCESS FACTORS TO APPLY ML IN SAS?

To answer RQ4, we derived the answer from data field strength (IF11) and limitation (IF12). We found that most of the studies (79%) explicitly described the strength of the proposed approach with little or without explanation of the limitations.

The strengths of the studies are frequently reported. For the strength of ML approaches, most of the researchers claimed that ML had the ability to improve the performance and interoperability by predicting the environment or behavior [17], [94]. Other studies argued that using ML approach could allow for flexible enhancement. The researchers with energy domain applications show that ML gives advantages in optimization by providing dynamic configuration.

ML is also proven to have ability to reduce the cost of self-adaptation. Reducing the cost is important in SAS because process of changing artifacts or behaviors in the runtime can be costly. The application of ML could also advantageously reduce adaptation costs by decreasing the number of reconfigurations [70]. The work proposed by Lu and Cukic in [62] shows that the ML approach works better when used to track the system's quality. Most of the works mentioned that ML allows for better performance in handling uncertainty by understanding the context variability. With the ability to learn from past behavior patterns, ML can avoid unnecessary adaptations so to make the adaptation mechanism cheaper and faster.

Another way of machine learning for handling uncertainty is providing probabilistic method and decision theory to assess the consequences of uncertainty that covers design time and runtime and also different sources of uncertainty.

Even though various reported strengths, there is very insufficient information on the limitations of ML in SAS. One of the reported limitations is the greater delays in comparison to other approaches [2]. Another work mentioned there is still a need to define the knowledge manually, which could be difficult to understand and manage in the larger system [12]. Another limitation of using an ML technique include the characteristics of ML itself. Therefore, it is important to understand the needs of specific attributes and required data for the learning purpose.

V. DISCUSSION

The discussion section starts with discussing the findings of the review. Next, it provides recommendations for future research directions of machine learning in SAS.

A. FINDINGS

The main finding for the research includes the following:

1) THE CONTRIBUTION OF MACHINE LEARNING APPLICATION TRENDS IN SAS

While analyzing the trends (IF02 and IF03), we found that the application of machine learning techniques in SAS has increased in the last 19 years. One of the contributing factors may be the increasing trend in using machine learning techniques in the software engineering domain. Machine learning is typically used to discover software patterns and to detect software defects. Furthermore, the increasing complexity of software systems also contributes to the trend. The size and costs triggered studies of a better algorithm efficient methods to optimize the system. Therefore, machine learning is a potentially remarkable solution to address these issues.

Based on the results of machine learning implementation in the primary studies, we found machine learning highly reactive, which allows the system to change behavior according to the current situation. Moreover, machine learning supports a better decision-making process by actively analyzing and learning the gathered context information, historical data, and policies. Machine learning is able to address the uncertainty in SAS by learning new adaptation rules dynamically and modifying the existing rules. The work proposed by Chen *et. al.* in [90] shows that machine learning can reduce adaptation costs while maintaining remarkable performance.

2) INADEQUATE JUSTIFICATION METHOD

One of the key insights derived from this survey is that there is a lack of justification for choosing machine learning techniques in the adaptation process. The extracted data results for IF10 show that the majority of the papers rely on the case study without proper justification for choosing a technique. Most of the primary studies ignore their domain characteristics and available knowledge. The techniques were chosen based on a limited understanding of machine learning characteristics and training data requirements. However, it is unwise to select the technique solely by evaluating the ability of the selected technique. Without proper justification and validation, the performance of the adaptation mechanism could suffer at run-time, due to an over-fitting issue, and the lack of the model's availability in the representation of the pattern behavior.

3) THE LACK OF PRACTICAL RUN-TIME VERIFICATION AND VALIDATION

We performed two validation methods for the filtered studies which are validation 1 (cross-checking agreement) and validation 2 (independent expert checking). The result of each step is summarized in Table 4.

Another finding of the research is the lack of practicality in run-time verification and validation. Improving the self-adaptation process only via adaptation logic is not adequate. This finding aligns with the limited studies which provide dynamic model verification for their adaptation process. There is a limitation of using machine learning in SAS is the risk of hurting the run-time performance due to the needs of practicality in runtime verification and validation. Therefore, current studies mostly use the combination of off-line model verification and on-line behavior adjustment through transfer learning. The implementation of the machine learning technique ensures the quality assessment of software systems throughout their entire life cycles.

B. FUTURE RESEARCH DIRECTIONS

Based on findings derived from the analysis, we identified research directions to optimize the use of machine learning approach in SAS.

1) TAXONOMY OF MACHINE LEARNING IN SAS

Machine learning techniques can complement the control loop and the engineering process in SAS. However, each machine learning technique has its own characteristic based on the need for domain knowledge and its advantages. Each technique can be used differently based on its characteristics. Unfortunately, the machine learning technique is usually chosen due to its ability. For example, the support vector machine becomes the most used technique because of its high accuracy results using its kernel functions. It usually provides the best prediction in comparison to the other machine learning classifier. Guidelines are needed for adopting the machine learning technique to improve the adaptation process in SAS. Therefore, our first recommendation for research direction is the need for a taxonomy that should provide a mechanism to choose the technique based on domain problem and machine learning characteristics. Based on the analyzed data items, we proposed an initial taxonomy for choosing ML technique in SAS as seen in Figure 6.

FIGURE 6. The proposed initial taxonomy for choosing machine learning technique in SAS.

Understanding the adaptation concern and purpose of adaptation is important to select the machine learning method due to different characteristics the aspects. The adaptation concern is divided into five levels based on the survey proposed by Weyns *et al.*in [89] such as architecture, behavior, framework, model, and verification. Meanwhile, the purpose of adaptation is divided into model checking, modeling, and reasoning. This aspect is related to the adaptation time that is categorized as proactive adaptation and reactive adaptation. The next aspect that should be considered is the decision criteria. This aspect categorized the machine learning technique based on its learning processes such as model-based method, goal-based method, and rule-based method.

The domain problem includes data collection, goal identification, and required monitored data for the learning process. The identifying domain problem is significant because of the characteristic of the machine learning task, which is usually problem-specific and formalism-dependent. Therefore, the attributes and features should be clearly identified. The next consideration is understanding the characteristics of the technique. The characteristics should include the need for domain knowledge, the size of training data, advantages, and disadvantages. Based on the characteristic, we know the need of a specific number of data and the objective of the machine learning technique can be different.

This factor is important for considering the adoption of a machine learning technique. Based on the proposed taxonomy, the choice of machine learning technique can be made. For example, if the monitored data is small and has a limited number of features, and the system has 'AS FAST AS POSSIBLE', then the support vector machine can be used. It is because SVM does not require domain knowledge and can work quickly with the least number of features.

2) AUTOMATIC MACHINE LEARNING TECHNIQUE SELECTION

From survey findings, we can see how the machine learning technique can be used in a software system to make them adaptive and self-configuring. The adaptation process of the self-adaptive support system can be achieved effectively by configuring the system to learn to adapt rather than make it behave like a control mode. However, choosing the machine learning technique is not a trivial task. This issue can raise a problem when the engineer does not have an adequate understanding of machine learning techniques.

Therefore, the second recommendation is to have an automatic process to choose a machine learning technique for adaptation in the runtime process. The aim is to minimize the potential disadvantages of choosing the wrong technique. The most important factor is to avoid conflict between characteristics of the software system's domain problem and the selected machine learning technique.

One of the keys of SAS is that the system should be able to adjust its behavior automatically in response to changes in its environments. The work in [18] argues that uncertainty is one of the main challenges in understanding the nature of the system such as behavior pattern and size of domain knowledge. It is hard to predict how the system may change over time. By automating machine learning selection, we can avoid these issues. That method will enhance the adaptation process by adjusting the adaptation model or rule.

3) DEEP LEARNING APPROACH

By analyzing the limitations of the primary studies, we found that machine learning requires more computational time than the other techniques. It is because machine learning is a proactive approach to learn and predict the environments. This practice can result in the delay of the adaptation process. This could pose a significant problem when dealing with complex systems with broad ranges of data.

This issue can be addressed by utilizing deep learning approaches. It is one of the branches of machine learning that has the ability to learn based on high-level data abstraction. Deep learning in software engineering has been heavily studied during the past years.

The work proposed by Yang *et al.* in [93] argued that deep learning has made significant improvements in speech recognition, visual object recognition, object detection, and many other domains, including drug discovery and genomics. The authors mention deep learning showed remarkable results in various tasks in natural language understanding, particularly topic classification, sentiment analysis, question answering, and language translation.

Deep learning architecture consists of multi-layered stacks of simple modules, including learning knowledge. These modules can be composed with the nonlinear input-output mapping. This characteristic gives deep learning the ability to support various knowledge representations. The usual approach for handling uncertainty in SAS is to utilize the knowledge representation and complex reasoning. Deep learning can be a new paradigm that replaces rule-based reasoning to address adaptation challenges in SAS.

Some studies of self-adaptation using deep learning have been studied in the area of network communication. One of application of deep learning in adaptive system is proposed by Papamartzivanos *et al.* in [103]. This study shows that deep learning is able to understand the nature of attacks based on reconstruction of environment data.

VI. LIMITATIONS OF THE STUDY

We conducted a systematic literature review about the application of ML technique in SAS on 78 primary studies published between 2001 until 2019. The results of this survey may have been affected with the coverage of search strategy, researcher's bias, imbalance publication venues, and inaccuracy of data extraction. These have been addressed as below.

A. INCOMPLETENESS OF SEARCH RESULTS

The search processes were organized both automatically and manually. The automatic searching resulted in hundreds of data points. The manual search process was done based on the title, keyword and abstract, and introduction. A single researcher selected the possible studies. There may be a case where relevant studies were not included in the search result. As a result, this literature review may not cover the entire literature. Therefore, this survey is only valid based on the 78 primary studies used in this systematic literature review. In order to reduce the possibility of missing primary studies not indexed in the selected data source, we also utilized Google Scholar to get the relevant studies.

B. RESEARCHERS' BIASES

Another threat to validity is related to the possibility of bias among the researchers. To reduce bias, we performed a cross-check analysis between main researchers which consist of one Professor and one PhD researcher. The researchers were asked to give feedback and supporting analyses for the other researchers. The results were compared and discussed to solve the conflict that may arise during the cross-check analysis.

C. IMBALANCES OF PRIMARY STUDIES DISTRIBUTION OVER PUBLICATION VENUE

As seen in Appendix Table 15, the publication venues of referenced works vary in journal to conference to workshop. Instead of choosing a specific venue, we gathered any paper that satisfied the inclusion and exclusion criteria. As the result, the publication venues of our primary studies reflect areas of artificial intelligence, software engineering, and cyber-physical systems. The primary studies were gathered from the significant part of the literature on ML in self-adaptive. Therefore, we can argue that the result of the literature review is valid.

D. INACCURACY OF DATA ITEM EXTRACTION

To address this issue, we adopt some categories from the existing SAS taxonomy. We also asked an independent researcher to validate the extracted data item. From the validation process, the independent researcher agreed with 91% of our extracted data item. We can ensure the accuracy of the study by providing the detail searching scope, data item and criteria for answering the research questions.

VII. CONCLUSION

The objective of this literature review was to assess the state-of-the-art implementation of ML approaches in handling self-adaptation. This work provided a detailed systematic literature review ML approaches proposed by 78 papers in the literature. In the existing literature, some numbers of SAS have been implemented based on ML methodology. The quantitative results were analyzed and presented, providing insight into trends of ML application in SAS.

TABLE 14. List of primary studies published from 2001 – 2019.

INDEX	TITLE	REF	PUBLISHER	TYPE	YEAR
PS01	Adaptive Agent Based System for State Estimation Using Dynamic Multidimensional Information Sources	[81]	Springer	conference	2001
PS02	ISAM, a software architecture for adaptive and distributed mobile applications	[4]	IEEE	conference	2002
PS03	Self-organising networks in modelling experimental data in software engineering	[69]	IET	journal	2002
PS04	Modeling, simulation, and optimization software framework for dynamic systems	[86]	IEEE	conference	2003
PS05	Towards adaptive soft computing-based software effort prediction	[77]	IEEE	conference	2004
PS06	A Fuzzy Evolutionary System for Concept Formation and Adaptive Behaviour in Software Agents	[58]	IEEE	conference	2005
PS07	Organizing and Visualizing Software Repositories Using the Growing Hierarchical Self-organizing Map	[84]	ACM	conference	2005
PS08	Hybrid Prediction Model for improving Reliability in Self-Healing System	[98]	IEEE	conference	2006
PS09	SHAGE: A Framework for Self-managed Robot Software	[110]	ACM	conference	2006
PS10	A real-time adaptive control of autonomic computing environments	[80]	ACM	conference	2007
PS11	Digitally Evolving Models for Dynamically Adaptive Systems	[33]	IEEE	conference	2007
PS12	Adaptive Action Selection in Autonomic Software Using Reinforcement Learning	[1]	IEEE	conference	2008
PS13	Video Agent: Interactive Autonomous Agents Generated from Real-world Creatures	[50]	ACM	conference	2008
PS14	Reinforcement learning-based dynamic adaptation planning method for architecture-based self-managed software	[49]	IEEE	conference	2009
PS15	The Design of Intelligent Security Defensive Software Based on Autonomic Computing	[78]	IEEE	conference	2009
PS16	A Software Self-Organizing Middleware for Smart Spaces Based on Fuzzy Logic	[34]	IEEE	conference	2010
PS17	Adaptation Issues in Software Architectures of Remote Health Care Systems	[61]	ACM	conference	2010
PS18	Toward a Fuzzy Control-based Approach to Design of Self-adaptive Software	[93]	ACM	conference	2010
PS19	Towards an intelligent security defense software's self-decision system	[46]	IEEE	conference	2010
PS20	Consequence oriented self-healing and autonomous diagnosis for highly reliable systems and software	[22]	IEEE	journal	2011
PS21	Evolutionary environmental modelling in self-managing software systems	[29]	IEEE	conference	2011
PS22	Fuzzy Control-Based Software Self-Adaptation: A Case Study in Mission Critical Systems	[94]	IEEE	conference	2011
PS23	An adaptive approach with active learning in software fault prediction	[62]	ACM	conference	2012
PS24	Runtime enforcement of information flow security in tree manipulating processes.	[52]	Springer	conference	2012
PS25	Towards a general supporting framework for self-adaptive software systems	[85]	IEEE	conference	2012
PS26	A learning-based framework for engineering feature-oriented self-adaptive software systems	[27]	IEEE	journal	2013
PS27	Dynamic decision networks for decision-making in self-adaptive systems: a case study	[7]	IEEE	conference	2013
PS28	Run-time adaptation of mobile applications using genetic algorithms	[70]	IEEE	conference	2013
PS29	Self-adaptive and sensitivity-aware qos modeling for the cloud	[14]	IEEE	conference	2013
PS30	A prediction-driven adaptation approach for self-adaptive sensor networks	[2]	ACM	conference	2014
PS31	An approach to clustering feature model based on adaptive behavior for dynamic software product line	[9]	IEEE	conference	2014
PS32	Autonomic resource provisioning for cloud-based software	[43]	ACM	conference	2014
PS33	AutoRELAX: automatically RELAXing a goal model to address uncertainty	[30]	Springer	journal	2014
PS34	Extending uml for the modeling of fuzzy self-adaptive software systems	[37]	IEEE	conference	2014
PS35	Fuaet: a tool for developing fuzzy self-adaptive software systems	[95]	ACM	conference	2014
PS36	Self-adaptation through incremental generative model transformations at runtime	[12]	ACM	conference	2014
PS37	Towards run-time adaptation of test cases for self-adaptive systems in the face of uncertainty	[31]	ACM	conference	2014
PS38	Adaptive knowledge bases in self-adaptive system design	[51]	IEEE	conference	2015
PS39	An agent-based self-adaptive mechanism with reinforcement learning	[99]	IEEE	conference	2015
PS40	Continuous collaboration: a case study on the development of an adaptive cyber-physical system	[40]	IEEE	conference	2015

TABLE 14. ***(Continued.)*** **List of primary studies published from 2001 – 2019.**

INDEX	TITLE	REF	PUBLISHER	TYPE	YEAR
PS41	Incorporating human intention into self-adaptive systems	[42]	IEEE	conference	2015
PS42	Model-based reinforcement learning approach for planning in self-adaptive software system	[39]	ACM	conference	2015
PS43	Sass: self-adaptation using stochastic search	[19]	IEEE	conference	2015
PS44	A generative genetic algorithm for evolving adaptation rules of software systems	[60]	ACM	conference	2016
PS45	A runtime framework for machine-augmented software design using unsupervised self-learning	[74]	IEEE	conference	2016
PS46	Handling uncertainty in self-adaptive software using self-learning fuzzy neural network	[36]	IEEE	conference	2016
PS47	Modeling self-adaptive software systems with learning petri nets	[24]	IEEE	journal	2016
PS48	Predicting maintainability of autonomic software systems using fuzzy logic	[56]	IEEE	conference	2016
PS49	Privacy dynamics: learning privacy norms for social software	[11]	ACM	conference	2016
PS50	Recognizing voice-based requirements to drive self-adaptive software systems	[101]	IEEE	conference	2016
PS51	A reinforcement learning-based framework for the generation and evolution of adaptation rules	[102]	IEEE	conference	2017
PS52	A web service-based approach for developing self-adaptive systems	[47]	Elsevier	journal	2017
PS53	Autonomous learning multi-model systems from data streams	[3]	IEEE	journal	2017
PS54	Combining machine-learning with invariants assurance techniques for autonomous systems	[63]	IEEE	conference	2017
PS55	Fiot: an agent-based framework for self-adaptive and self-organizing applications based on the internet of things	[68]	Elsevier	journal	2017
PS56	Self-adaptive and online qos modeling for cloud-based software services	[15]	IEEE	journal	2017
PS57	Self-adaptive dynamic decision making processes	[5]	IEEE	conference	2017
PS58	Using a multi-agent system and artificial intelligence for monitoring and improving the cloud performance and security	[35]	Elsevier	journal	2017
PS59	A concept for proactive knowledge construction in self-learning autonomous systems	[82]	IEEE	conference	2018
PS60	A learning approach to enhance assurances for real-time self-adaptive systems	[72]	IEEE	conference	2018
PS61	A self-learning approach for validation of runtime adaptation in service-oriented systems	[67]	Springer	journal	2018
PS62	An adaptive decision-making method with fuzzy bayesian reinforcement learning for robot soccer	[79]	Elsevier	journal	2018
PS63	Framework for building self-adaptive component applications based on reinforcement learning	[6]	IEEE	conference	2018
PS64	Sacre: supporting contextual requirements' adaptation in modern self-adaptive systems in the presence of uncertainty at runtime	[100]	Elsevier	journal	2018
PS65	Satisfy: towards a self-learning analyzer for time series forecasting in self-improving systems	[53]	IEEE	conference	2018
PS66	Managing uncertainty in self-adaptive systems with plan reuse and stochastic search.	[111]	ACM	conference	2018
PS67	To adapt or not to adapt? Technical debt and learning driven self-adaptation for managing runtime performance	[16]	ACM	conference	2018
PS68	Training prediction models for rule-based self-adaptive systems	[32]	IEEE	conference	2018
PS69	Using reinforcement learning to handle the runtime uncertainties in self-adaptive software	[91]	Springer	conference	2018
PS70	Learning based adaptation framework for elastic software systems	[83]	N/A	conference	2019
PS71	Efficient Analysis of Large Adaptation Spaces in Self-Adaptive Systems using Machine Learning	[71]	ACM	conference	2019
PS72	Machine learning meets quantitative planning: Enabling self-adaptation in autonomous robots	[44]	ACM	conference	2019
PS73	Enhancing context specifications for dependable adaptive systems: A data mining approach	[73]	Elsevier	journal	2019
PS74	A Machine Learning-Driven Approach for Proactive Decision Making in Adaptive Architectures	[66]	IEEE	conference	2019
PS75	Reasoning Non-Functional Requirements Trade-off in Self-Adaptive Systems Using Multi-Entity Bayesian Network Modeling	[76]	KSCI	journal	2019
PS76	Machine learning to guide performance testing: An autonomous test framework	[65]	IEEE	conference	2019
PS77	Self-optimizing and self-programming computing systems: a combined compiler, complex networks, and machine learning approach	[92]	IEEE	journal	2019
PS78	Self-learning and self-adaptive resource allocation for cloud-based software services	[17]	Wiley	journal	2019

TABLE 15. Distribution of primary studies over publication venues.

index	Publication source	Frequency	Primary Studies
SR01	American Control Conference	1	PS04
SR02	Annual Computer Software and Applications Conference	5	PS22, PS25, PS39, PS46, PS50
SR03	Annual Meeting of the Fuzzy Information	1	PS05
SR04	Asia-Pacific Symposium on Internetware	3	PS18, PS35, PS45
SR05	Chinese Control and Decision Conference	1	PS34
SR06	Computers \& Electrical Engineering	1	PS52
SR07	Concurrency and Computation: Practice and Experience	1	PS78
SR08	Conference of the center for advanced studies on Collaborative research	1	PS10
SR09	Developments in E-systems Engineering	1	PS21
SR10	Emperical Software Engineering	1	PS33
SR11	Euromicro Conference on Software Engineering and Advanced Applications	1	PS38
SR12	Expert Systems with Applications	1	PS64
SR13	Federation of International Conferences on Software Technologies: Applications and Foundations	1	PS69
SR14	Future Generation Computer Systems	1	PS58
SR15	IEE Proceedings-Computers and Digital Techniques	1	PS03
SR16	IEEE Conference on Cognitive and Computational Aspects of Situation Management	1	PS57
SR17	IEEE Transactions on Fuzzy Systems	1	PS53
SR18	IEEE Transactions on Reliability	1	PS20
SR19	IEEE transactions on software engineering	2	PS26, PS56
SR20	IEEE Transactions on Systems, Man, and Cybernetics: Systems	1	PS47
SR21	IEEE Transactions on Very Large Scale Integration Systems	1	PS77
SR22	Information Sciences	2	PS55, PS62
SR23	Information and Software Technology	1	PS73
SR24	International Conference on Autonomic and Autonomous Systems	1	PS12
SR25	International Conference on Autonomic Computing	3	PS46, PS51, PS68
SR26	International Conference on Computing, Communication and Automation	1	PS48
SR27	International Conference on Fuzzy Systems	1	PS06
SR28	International Conference on High Performance Computing and Communications	1	PS16
SR29	International Conference on Information Management and Engineering	1	PS19
SR30	International Conference on Information Science \& Applications	1	PS31
SR31	International Conference on Intelligent Computation Technology and Automation	1	PS15
SR32	International Conference on Performance Engineering	1	PS67
SR33	International Conference on Predictive Models in Software Engineering	1	PS23
SR34	International Conference on Services Computing	1	PS63
SR35	International Conference on Software Architecture Companion	1	PS74
SR36	International Conference on Software Engineering	2	PS36, PS41
SR37	International Conference on Software Engineering and Knowledge Engineering	1	PS70
SR38	International Conference on Software Engineering Companion	1	PS53
SR39	International Conference on Software Engineering Research, Management and Applications	1	PS08
SR40	International Conference on Software Testing, Verification and Validation Workshops	1	PS76
SR41	International Conference on Ubiquitous Information Management and Communication	1	PS42
SR42	International Symposium on Computers and Communications	1	PS02
SR43	International Symposium on Engineering Secure Software and Systems	1	PS24
SR44	International symposium on software engineering for adaptive and self-managing systems	12	PS27, PS28, PS29, PS30, PS32, PS37, PS43, PS49, PS60, PS66, PS71, PS72
SR45	International workshop on Self-adaptation and self-managing systems	1	PS09
SR46	International Workshop on Self-Adaptive Software	1	PS01
SR47	International Workshop on Software Engineering for Adaptive and Self-Managing Systems	2	PS11, PS14
SR48	International Workshop on Software Engineering for Smart Cyber-Physical Systems	1	PS40
SR49	International Workshops on Foundations and Applications of Self* Systems	2	PS59, PS65
SR50	Journal of The Korea Society of Computer and Information	1	PS75
SR51	Service Oriented Computing and Applications	1	PS61
SR52	Symposium on Applied computing	1	PS07
SR53	Symposium on Virtual reality software and technology	1	PS13
SR54	Workshop on Software Engineering in Health Care	1	PS17

The existing work in the area of SAS shows the implementation of ML approach can improve the performance of the system. We can see how the ML technique used in a software system can make systems adaptive and self-configuring. The adaptation process in SAS can be achieved effectively by making the system learn to adapt rather than make it behave like a control mode. This review illustrates that ML can significantly contribute to SAS by providing a method to understand the system's behavior and environments. The primary studies cited with various domain applications show that ML provides advantages in optimization through providing dynamic configuration.

The limited literature on ML in SAS indicates there is much room for improvement and future work in this area. Furthermore, only a few studies were identified that used a systematic justification for choosing the ML approach. Therefore, we provide recommendations that could be useful in the software engineering community in the next steps to improve research in the SAS field.

APPENDIX

See Table 14 and 15 here.

REFERENCES

[1] M. Amoui, M. Salehie, S. Mirarab, and L. Tahvildari, "Adaptive action selection in autonomic software using reinforcement learning," in *Proc. 4th Int. Conf. Autonomic Auto. Syst. (ICAS)*, Mar. 2008, pp. 175–181.

[2] I. D. P. Anaya, V. Simko, J. Bourcier, N. Plouzeau, and J.-M. Jézéquel, "A prediction-driven adaptation approach for self-adaptive sensor networks," in *Proc. 9th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst.*, 2014, pp. 145–154.

[3] P. P. Angelov, X. Gu, and J. C. Principe, "Autonomous learning multimodel systems from data streams," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 2213–2224, Aug. 2018.

[4] L. Augustin, A. C. Yarnin, J. L. V. Barbosa, and C. F. R. Geyer, "ISAM, a software architecture for adaptive and distributed mobile applications," in *Proc. ISCC 7th Int. Symp. Comput. Commun.*, 2002, pp. 333–338.

[5] K. Baclawski, E. S. Chan, D. Gawlick, A. Ghoneimy, K. C. Gross, and Z. H. Liu, "Self-adaptive dynamic decision making processes," in *Proc. IEEE Conf. Cognit. Comput. Aspects Situation Manage. (CogSIMA)*, Mar. 2017, pp. 1–6.

[6] N. Belhaj, D. Belaid, and H. Mukhtar, "Framework for building self-adaptive component applications based on reinforcement learning," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2018, pp. 17–24.

[7] N. Bencomo, A. Belaggoun, and V. Issarny, "Dynamic decision networks for decision-making in self-adaptive systems: A case study," in *Proc. 8th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst. (SEAMS)*, May 2013, pp. 113–122.

[8] A. Berns and S. Ghosh, "Dissecting Self properties," in *Proc. 3rd IEEE Int. Conf. Self-Adapt. Self-Organizing Syst.*, Sep. 2009, pp. 10–19.

[9] P. Boonon and P. Muenchaisri, "An approach to clustering feature model based on adaptive behavior for dynamic software product line," in *Proc. Int. Conf. Inf. Sci. Appl. (ICISA)*, May 2014, pp. 1–4.

[10] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, 2007.

[11] G. Calikli, M. Law, A. K. Bandara, A. Russo, L. Dickens, B. A. Price, A. Stuart, M. Levine, and B. Nuseibeh, "Privacy dynamics: Learning privacy norms for social software," in *Proc. 11th Int. Workshop Softw. Eng. Adapt. Self-Manag. Syst.*, 2016, pp. 47–56.

[12] B. Chen, X. Peng, Y. Yu, B. Nuseibeh, and W. Zhao, "Self-adaptation through incremental generative model transformations at runtime," in *Proc. 36th Int. Conf. Softw. Eng.*, 2014, pp. 676–687.

[13] L. Chen, M. A. Babar, and H. N. Zhang, "Towards an evidence-based understanding of electronic data sources," in *Proc. Conf.*, Apr. 2010, pp. 135–138.

[14] T. Chen and R. Bahsoon, "Self-adaptive and sensitivity-aware QoS modeling for the cloud," in *Proc. 8th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst. (SEAMS)*, May 2013, pp. 43–52.

[15] T. Chen and R. Bahsoon, "Self-adaptive and online qos modeling for cloud-based software services," *IEEE Trans. Softw. Eng.*, vol. 43, no. 5, pp. 453–475, May 2017.

[16] T. Chen, R. Bahsoon, S. Wang, and X. Yao, "To adapt or not to adapt?: Technical debt and learning driven self-adaptation for managing runtime performance," in *Proc. ACM/SPEC Int. Conf. Perform. Eng.*, Mar. 2018, pp. 48–55.

[17] X. Chen, J. Lin, B. Lin, T. Xiang, Y. Zhang, and G. Huang, "Self-learning and self-adaptive resource allocation for cloud-based software services," *Concurrency Comput., Pract. Exper.*, vol. 12, Dec. 2018, Art. no. e4463.

[18] B. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, "Software engineering for self-adaptive systems," in *Proc. LNCS*, 2019, p. 5525.

[19] Z. Coker, D. Garlan, and C. Le Goues, "Sass: Self-adaptation using stochastic search," in *Proc. 10th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst.*, 2018, pp. 168–174.

[20] *An Architectural Blueprint for Autonomic Computing*, IBM, Endicott, NY, USA, 2006, pp. 1–6.

[21] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of auvs with control input nonlinearities using reinforcement learning," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 6, pp. 1019–1029, Jan. 2017.

[22] Y. Dai, Y. Xiang, Y. Li, L. Xing, and G. Zhang, "Consequence oriented self-healing and autonomous diagnosis for highly reliable systems and software," *IEEE Trans. Rel.*, vol. 60, no. 2, pp. 369–380, Jun. 2011.

[23] S. De Lemos, *Software engineering for self-adaptive systems: A second research roadmap*. Cham, Switzerland: Springer, 2013, pp. 1–32.

[24] Z. Ding, Y. Zhou, and M. Zhou, "Modeling self-adaptive software systems with learning Petri nets," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 4, pp. 483–498, Apr. 2016.

[25] T. Dyb and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, nos. 9–10, pp. 833–859, Aug. 2008.

[26] A. Elkhodary, N. Esfahani, and S. Malek, "FUSION: A framework for engineering self-tuning self-adaptive software systems," in *Proc. 18th ACM SIGSOFT Int. Symp. Found. Softw. Eng.*, 2010, pp. 7–16.

[27] N. Esfahani, A. Elkhodary, and S. Malek, "A learning-based framework for engineering feature-oriented self-adaptive software systems," *IEEE Trans. Softw. Eng.*, vol. 39, no. 11, pp. 1467–1493, Nov. 2013.

[28] N. Esfahani and S. Malek, "Uncertainty in self-adaptive software systems," in *Proc. Softw. Eng. Self-Adapt. Syst.*, 2013, pp. 214–238.

[29] H. Forsyth, A. Laws, and A. Bendiab, "Evolutionary environmental modelling in self-managing software systems," in *Proc. Conf. Develop. Syst. Eng.*, 2011, pp. 370–375.

[30] E. M. Fredericks, B. DeVries, and B. H. C. Cheng, "AutoRELAX: Automatically RELAXing a goal model to address uncertainty," *Empirical Softw. Eng.*, vol. 19, no. 5, pp. 1466–1501, Oct. 2014.

[31] E. M. Fredericks, B. DeVries, and B. H. C. Cheng, "Towards run-time adaptation of test cases for self-adaptive systems in the face of uncertainty," in *Proc. 9th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst.*, 2014, pp. 17–26.

[32] S. Ghahremani, C. M. Adriano, and H. Giese, "Training prediction models for rule-based self-adaptive systems," in *Proc. IEEE Int. Conf. Autonomic Comput. (ICAC)*, Sep. 2018, pp. 187–192.

[33] H. J. Goldsby, D. B. Knoester, B. H. C. Cheng, P. K. McKinley, and C. A. Ofria, "Digitally evolving models for dynamically adaptive systems," in *Proc. Int. Workshop Softw. Eng. Adapt. Self-Manag. Syst. (SEAMS)*, May 2007, p. 13.

[34] C. Gouin-Vallerand, B. Abdulrazak, S. Giroux, and M. Mokhtari, "A software self-organizing middleware for smart spaces based on fuzzy logic," in *Proc. IEEE 12th Int. Conf. High Perform. Comput. Commun. (HPCC)*, Sep. 2010, pp. 138–145.

[35] D. Grzonka, A. Jakobik, J. Koodziej, and S. Pllana, "Using a multi-agent system and artificial intelligence for monitoring and improving the cloud performance and security," *Future Gener. Comput. Syst.*, vol. 86, pp. 1106–1117, Feb. 2018.

[36] D. Han, J. Xing, Q. Yang, J. Li, and H. Wang, "Handling uncertainty in self-adaptive software using self-learning fuzzy neural network," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jun. 2016, pp. 540–545.

[37] D. Han, Q. Yang, and J. Xing, "Extending UML for the modeling of fuzzy self-adaptive software systems," in *Proc. 26th Chin. Control Decision Conf. (CCDC)*, May 2014, pp. 2400–2406.

[38] J. A. Harer, "Automated software vulnerability detection with machine learning," 2018, *arXiv:1803.04497*. [Online]. Available: http://arxiv.org/abs/1803.04497

[39] H. Ho and E. Lee, "Model-based reinforcement learning approach for planning in self-adaptive software system," in *Proc. 9th Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2015, p. 103.

[40] M. Holzl and T. Gabor, "Continuous collaboration: A case study on the development of an adaptive cyber-physical system," in *Proc. IEEE/ACM 1st Int. Workshop Softw. Eng. for Smart Cyber-Phys. Syst.*, May 2015, pp. 19–25.

[41] J. Huang, Y.-F. Li, and M. Xie, "An empirical analysis of data preprocessing for machine learning-based software cost estimation," *Inf. Softw. Technol.*, vol. 67, pp. 108–127, Nov. 2015.

[42] S. Huang and P. Miranda, "Incorporating human intention into self-adaptive systems," in *Proc. IEEE/ACM 37th IEEE Int. Conf. Softw. Eng.*, May 2015, pp. 571–574.

[43] P. Jamshidi, A. Ahmad, and C. Pahl, "Autonomic resource provisioning for cloud-based software," in *Proc. 9th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst.*, 2014, pp. 95–104.

[44] P. Jamshidi, J. Camara, B. Schmerl, C. Kaestner, and D. Garlan, "Machine learning meets quantitative planning: Enabling self-adaptation in autonomous robots," in *Proc. IEEE/ACM 14th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst. (SEAMS)*, May 2019, pp. 39–50.

[45] N. Jennings, "On agent-based software engineering," *Artif. Intell.*, vol. 117, pp. 277–296, Dec. 2002.

[46] W. Jin-dong, S. Liu-qing, W. Na, and G. Qiang, "Towards an intelligent security defense software's self-decision system," in *Proc. 2nd IEEE Int. Conf. Inf. Manage. Eng.*, 2010, pp. 204–206.

[47] D. A. Kafaf and D.-K. Kim, "A Web service-based approach for developing self-adaptive systems," *Comput. Electr. Eng.*, vol. 63, pp. 260–276, Oct. 2017.

[48] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[49] D. Kim and S. Park, "Reinforcement learning-based dynamic adaptation planning method for architecture-based self-managed software," in *Proc. ICSE Workshop Softw. Eng. Adapt. Self-Manag. Syst.*, May 2009, pp. 76–85.

[50] Y. Kitamura, R. Rong, Y. Hirano, K. Asai, and F. Kishino, "Video agent: Interactive autonomous agents generated from real-world creatures," in *Proc. ACM Symp. Virtual reality Softw. Technol.*, 2008, pp. 30–38.

[51] V. Klos, T. Gothel, and S. Glesner, "Adaptive knowledge bases in self-adaptive system design," in *Proc. 41st Euromicro Conf. Softw. Eng. Adv. Appl.*, Aug. 2015, pp. 472–478.

[52] M. Kovács and H. Seidl, "Runtime enforcement of information flow security in tree manipulating processes," in *Proc. Int. Symp. Eng. Secure Softw. Syst.*, 2012, pp. 46–59.

[53] C. Krupitzer, M. Pfannemuller, J. Kaddour, and C. Becker, "SATISFy: Towards a self-learning analyzer for time series forecasting in self-improving systems," in *Proc. IEEE 3rd Int. Workshops Found. Appl. Self Syst. (FASW)*, Sep. 2018, pp. 182–189.

[54] C. Krupitzer, M. Pfannemüller, V. Voss, and C. Becker, "Comparison of approaches for developing self-adaptive systems," Inf. Syst. II, Univ. Mannheim, Mannheim, Germany, Tech. Rep., 2018. [Online]. Available: https://madoc.bib.uni-mannheim.de/43909/

[55] C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker, "A survey on engineering approaches for self-adaptive systems," *Pervas. Mobile Comput.*, vol. 17, pp. 184–206, Feb. 2015.

[56] A. Kumar and V. Tyagi, "Predicting maintainability of autonomic software systems using fuzzy logic," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, Apr. 2016, pp. 353–357.

[57] Z. Kurtanovic and W. Maalej, "Automatically classifying functional and non-functional requirements using supervised machine learning," in *Proc. IEEE 25th Int. Requirements Eng. Conf. (RE)*, Sep. 2017, pp. 490–495.

[58] E. Langham, "A fuzzy evolutionary system for concept formation and adaptive behaviour in software agents," in *Proc. 14th IEEE Int. Conf. Fuzzy Syst.*, 2005, pp. 631–635.

[59] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, p. 436, May 2015.

[60] Y. Liu, W. Zhang, and W. Jiao, "A generative genetic algorithm for evolving adaptation rules of software systems," in *Proc. 8th Asia–Pacific Symp. Internetware*, 2016, pp. 103–107.

[61] O. Loques and A. Sztajnberg, "Adaptation issues in software architectures of remote health care systems," in *Proc. ICSE Workshop Softw. Eng. Health Care*, 2010, pp. 24–28.

[62] H. Lu and B. Cukic, "An adaptive approach with active learning in software fault prediction," in *Proc. 8th Int. Conf. Predictive Models Softw. Eng.*, 2012, pp. 79–88.

[63] P. Mallozzi, "Combining machine-learning with invariants assurance techniques for autonomous systems," in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng. Companion (ICSE-C)*, May 2017, pp. 485–486.

[64] J. Mendel, *Uncertain rule-based fuzzy logic systems: Introduction and nework directions*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000, pp. 20–25.

[65] M. Helali Moghadam, M. Saadatmand, M. Borg, M. Bohlin, and B. Lisper, "Machine learning to guide performance testing: An autonomous test framework," in *Proc. IEEE Int. Conf. Softw. Test., Verification Validation Workshops (ICSTW)*, Apr. 2019, pp. 164–167.

[66] H. Muccini and K. Vaidhyanathan, "A machine learning-driven approach for proactive decision making in adaptive architectures," in *Proc. IEEE Int. Conf. Softw. Archit. Companion (ICSA-C)*, Mar. 2019, pp. 242–245.

[67] L. Mutanu and G. Kotonya, "A self-learning approach for validation of runtime adaptation in service-oriented systems," in *Proc. conf. Service Oriented Comput. Appl.*, 2018, pp. 1–14.

[68] N. M. do Nascimento and C. J. P. de Lucena, "FIoT: An agent-based framework for self-adaptive and self-organizing applications based on the Internet of Things," *Inf. Sci.*, vol. 378, pp. 161–176, Feb. 2017.

[69] S. K. Oh, W. Pedrycz, and H. S. Park, "Self-organising networks in modelling experimental data in software engineering," *IEE Proc. Comput. Digit. Techn.*, vol. 149, no. 3, pp. 61–78, May 2002.

[70] G. Pascual, M. Pinto, and L. Fuentes, "Run-time adaptation of mobile applications using genetic algorithms," *Proc. 8th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst.*, 2013, pp. 73–82.

[71] F. Quin, D. Weyns, T. Bamelis, S. Singh Buttar, and S. Michiels, "Efficient analysis of large adaptation spaces in self-adaptive systems using machine learning," in *Proc. IEEE/ACM 14th Int. Symp. Softw. Eng. for Adapt. Self-Manag. Syst. (SEAMS)*, May 2019, pp. 1–12.

[72] A. Rodrigues, R. D. Caldas, G. N. Rodrigues, T. Vogel, and P. Pelliccione, "A learning approach to enhance assurances for real-time self-adaptive systems," in *Proc. 13th Int. Conf. Softw. Eng. Adapt. Self-Manag. Syst.*, 2018, pp. 206–216.

[73] A. Rodrigues, G. N. Rodrigues, A. Knauss, R. Ali, and H. Andrade, "Enhancing context specifications for dependable adaptive systems: A data mining approach," *Inf. Softw. Technol.*, vol. 112, pp. 115–131, Aug. 2019.

[74] R. R. Filho and B. Porter, "A runtime framework for machine-augmented software design using unsupervised self-learning," in *Proc. IEEE Int. Conf. Autonomic Comput. (ICAC)*, Jul. 2016, pp. 231–232.

[75] Saeed, A.A.A., Lee, S.W., 2019, "Reasoning non-functional requirements trade-off in self-adaptive systems using multi-entity Bayesian network modeling," *J. The Korea Soc. Comput. Inf.*, vol. 24, pp. 65–75.

[76] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Trans. Auto. Adapt. Syst.*, vol. 4, no. 2, pp. 1–42, May 2009.

[77] M. O. Saliu, M. Ahmed, and J. AlGhamdi, "Towards adaptive soft computing based software effort prediction," in *Proc. IEEE Annu. Meeting Fuzzy Inf.*, 2004, pp. 16–21.

[78] L. Shen, J. Wang, K. Wang, and H. Zhang, "The design of intelligent security defensive software based on autonomic computing," in *Proc. 2nd Int. Conf. Intell. Comput. Technol. Autom.*, 2009, pp. 489–491.

[79] H. Shi, Z. Lin, S. Zhang, X. Li, and K.-S. Hwang, "An adaptive decision-making method with fuzzy Bayesian reinforcement learning for robot soccer," *Inf. Sci.*, vols. 436–437, pp. 268–281, Apr. 2018.

[80] B. Solomon, D. Ionescu, M. Litoiu, and M. Mihaescu, "A real-time adaptive control of autonomic computing environments," in *Proc. Conf. Center Adv. Stud. Collaborative Res.*, 2007, pp. 124–136.

[81] A. Soto and P. Khosla, "Adaptive agent based system for state estimation using dynamic multidimensional information sources," in *Proc. Int. Workshop Self-Adapt. Softw.*, 2001, pp. 66–83.

[82] A. Stein, S. Tomforde, A. Diaconescu, J. Hähner, C. Müller-Schloer, "A concept for proactive knowledge construction in self-learning autonomous systems," in *Proc. IEEE 3rd Int. Workshops Found. Appl. Self Syst.*, May 2018, pp. 204–213.

[83] Y. Sun, X. Cai, and K. Loparo, "Learning–based adaptation framework for elastic software systems," in *Proc. 31st Int. Conf. Softw. Eng. Knowl. Eng.*, Jul. 2019, pp. 281–286.

[84] S. Tangsripairoj and M. H. Samadzadeh, "Organizing and visualizing software repositories using the growing hierarchical self-organizing map," in *Proc. ACM Symp. Appl. Comput.*, 2005, pp. 1539–1545.

[85] L. Wang, Y. Gao, C. Cao, and L. Wang, "Towards a general supporting framework for self-adaptive software systems," in *Proc. IEEE 36th Annu. Comput. Softw. Appl. Conf. Workshops*, Jul. 2012, pp. 158–163.

[86] D. Webb and W. F. Ramirez, "Modeling, simulation, and optimization software framework for dynamic systems," in *Proc. Amer. Control Conf.*, 2003, pp. 2395–2400.

[87] G. Weiss, "Multiagent systems: A modern approach to distributed artificial intelligence," *Int. J. Comput. Intell. Appl.*, vol. 1, pp. 331–334, Feb. 2011.

[88] D. Weyns, "Software engineering of self-adaptive systems," in *Handbook Software Engineering*. Cham, Switzerland: Springer, 2019, pp. 399–443.

[89] D. Weyns, M. U. Iftikhar, S. Malek, and J. Andersson, "Claims and supporting evidence for self-adaptive systems: A literature study," in *Proc. 7th Int. Symp. Softw. Eng. for Adapt. Self-Manag. Syst. (SEAMS)*, Jun. 2012, pp. 89–98.

[90] D. Weyns, M. U. Iftikhar, D. G. de la Iglesia, and T. Ahmad, "A survey of formal methods in self-adaptive systems," in *Proc. 5th Int. Conf. Comput. Sci. Softw. Eng.*, 2012, pp. 66–79.

[91] T. Wu, Q. Li, L. Wang, L. He, and Y. Li, "Using reinforcement learning to handle the runtime uncertainties in self-adaptive software," in *Proc. Int. Conf. Softw. Technol., Appl. Found.* Cham, Switzerland: Springer, 2018, pp. 387–393.

[92] Y. Xiao, S. Nazarian, and P. Bogdan, "Self-optimizing and self programming computing systems: A combined compiler, complex networks, and machine learning approach," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 6, pp. 1416–1427, Jun. 2019.

[93] Q. Yang, J. Lü, J. Li, X. Ma, W. Song, and Y. Zou, "Toward a fuzzy control-based approach to design of self-adaptive software," in *Proc. 2nd Asia–Pacific Symp. Internetware*, 2010, p. 15.

[94] Q. Yang, J. Lu, J. Xing, X. Tao, H. Hu, and Y. Zou, "Fuzzy control-based software self-adaptation: A case study in mission critical systems," in *Proc. IEEE 35th Annu. Comput. Softw. Appl. Conf. Workshops*, Jul. 2011, pp. 13–18.

[95] Q. Yang, X. Tao, H. Xie, J. Xing, and W. Song, "FuAET: A tool for developing fuzzy self-adaptive software systems," in *Proc. 6th Asia–Pacific Symp. Internetware*, 2014, pp. 54–63.

[96] Z. Yang, Z. Li, Z. Jin, and Y. Chen, "A systematic literature review of requirements modeling and analysis for self-adaptive systems," in *Proc. Int. Working Conf. Requirement Eng., Found. Softw. Qual.*, 2014, pp. 55–71.

[97] D. Ye, M. Zhang, and A. V. Vasilakos, "A survey of self-organization mechanisms in multiagent systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 3, pp. 441–461, Mar. 2017.

[98] G. Yoo, J. Park, and E. Lee, "Hybrid prediction model for improving reliability in self-healing system," in *Proc. 4th Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, 2006, pp. 108–116.

[99] D. Yu, Q. Li, L. Wang, and Y. Lin, "An agent-based self-adaptive mechanism with; Reinforcement learning," in *Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf.*, Jul. 2015, pp. 582–585.

[100] E. Zavala, X. Franch, J. Marco, A. Knauss, and D. Damian, "Sacre: Supporting contextual requirements' adaptation in modern self-adaptive systems in the presence of uncertainty at runtime," *Expert Syst. Appl.*, vol. 98, pp. 166–188, Dec. 2018.

[101] X. Zhang, Q. Yang, J. Xing, and D. Han, "Recognizing voice-based requirements to drive self-adaptive software systems," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jun. 2016, pp. 534–539.

[102] T. Zhao, W. Zhang, H. Zhao, and Z. Jin, "A reinforcement learning-based framework for the generation and evolution of adaptation rules," in *Proc. IEEE Int. Conf. Autonomic Comput. (ICAC)*, Jul. 2017, pp. 103–112.

[103] D. Papamartzivanos, F. Gomez Marmol, and G. Kambourakis, "Introducing deep learning self-adaptive misuse network intrusion detection systems," *IEEE Access*, vol. 7, pp. 13546–13560, 2019.

[104] Andersson, Jesper, "Software engineering processes for self-adaptive systems," in *Software Engineering for Self-Adaptive Systems*. Berlin, Germany: Springer, 2013, pp. 51–75.

[105] E. Yuan, N. Esfahani, and S. Malek, "A systematic survey of self-protecting software systems," *ACM Trans. Auto. Adapt. Syst.*, vol. 8, no. 4, pp. 1–41, Jan. 2014.

[106] C. Raibulet, "Towards a taxonomy for the evaluation of Self-software," in *Proc. IEEE 3rd Int. Workshops Found. Appl. Self Syst.*, Sep. 2018, pp. 22–23.

[107] C. Raibulet, "Hints on quality evaluation of self-systems," in *Proc. IEEE 8th Int. Conf. Self-Adapt. Self-Organizing Syst.*, Sep. 2014, pp. 185–186.

[108] Y. Brun, "Engineering self-adaptive systems through feedback loops," in *Software engineering for self-adaptive systems*. Berlin, Germany: Springer, 2009, pp. 48–70.

[109] S. Dobson, "A survey of autonomic communications," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, pp. 223–259, 2006.

[110] D. Kim *et al.*, "SHAGE: A framework for self-managed robot software," in *Proc. Int. Workshop Self-Adaptation Self-Manag. Syst.*, 2006.

[111] C. Kinneer *et al.*, "Managing uncertainty in self-adaptive systems with plan reuse and stochastic search," in *Proc. 13th Int. Conf. Softw. Eng. Adapt. Self-Manag. Syst.*, 2018.

THERESIA RATIH DEWI SAPUTRI received the B.S. degree in information system from the Sepuluh Nopember Institute of Technology (ITS), Surabaya, Indonesia, in 2011, the M.S. degree in computer science and engineering from Ajou University, Suwon, South Korea, in 2014, and the Ph.D. degree, in 2020. She is currently a Lecturer with the Department of Informatics, Ciputra University, Indonesia. Her research interests include the development of sustainable software systems, requirements engineering for software sustainability, machine learning, and self-adaptive systems.

SEOK-WON LEE (Senior Member, IEEE) is currently a Full Professor with the Department of Software and Computer Engineering and the Chair of the Department of Artificial Intelligence, Ajou University, South Korea. He was a Faculty Member with The University of Texas at San Antonio and the University of North Carolina at Charlotte, USA. He also worked with Science Applications International Corporation (SAIC) and the IBM T. J. Watson Research Center as a Senior Research Scientist. He has published more than 180 peer-reviewed articles. His research interests include software engineering, knowledge acquisition, machine learning, and information assurance. He is also a Senior Member ACM and AAAI.